Asian Journal of Engineering and Applied Technology
ISSN: 2249-068X (P) Vol. 8 No. 2, 2019, pp. 34-36

DOI: https://doi.org/10.51983/ajeat-2019.8.2.1142

Modified Form of Cayley Hash Function

V. Vibitha Kochamani[1] and P. L. Lilly[2]
[1&2]Department of Mathematics, St. Joseph's College (Autonomous), Irinjalakuda, Kerala, India
E-Mail: myvkumari@gmail.com, sr.christy@gmail.com
(Received 1 March 2019; Revised 30 March 2019; Accepted 30 April 2019; Available online 7 May 2019)

Abstract **- In 1994 (AT CRYPTO94) introduced the celebrated Zemor-Tillich hash function over $SL_2(F_{2^n})$ is mathematically very efficient and simple method but now finally it was broken by Grassl *et al.*, 2011. Yet with a new choice of generators Zemor-Tillich constructions still remains of interest and a lot of construction was based on this type of hash function was created. One of our new construction is the devised Hash Function as follows: to an arbitrary text of {0, 1} *, associate the string of {A, B} obtained by substituting 0 for A and 1 for B, then assign to A and B values of adequately chosen matrices of Heis(Z). Now, in this paper we suggest a new version of a Cayley hash function using a discrete Heisenberg group. The Hashed value is the computed product. We improved the security Properties of the Cayley Hash Function. Here we hold a different concept to form a Factorisation Problem harder. We hold an efficient way to impose limits on the type of factorisations for attacking H.**
Keywords: **Cayley Hash Function, Discrete Heisenberg Group, Cayley Graph, Hash Function**

I. INTRODUCTION

Cryptographic hash functions play a vital role in modern cryptography. Hash functions [10, and 11] are simple and easy-to compute, that takes a variable length input and converts it to a fixed-length output. If such a function satisfies additional requirements it can be used for cryptographic applications, for example to protect the authenticity of messages sent over an insecure channel. The basic idea is that the hash result provides a unique imprint of a message, and that the protection of a short imprint is easier than the protection of message itself.

A cryptographic hash function can provide assurance of data integrity. Hash functions are widely used in numerous cryptographic protocols and a lot of work has already been put into devising adequate hashing schemes. Hash functions are used as compact representations or digital finger prints, of data and to provide message integrity. Some hash functions like: MD5, HAVAL, SHA. It was proved that these hash functions are no longer secure in current use and have been shown to be vulnerable. Early suggestions (particularly SHA family) did not really use any mathematical ideas apart from Merkle-Damgard [3,14] construction for producing collision resistant hash functions from collision resistant compression functions, the main idea was just to "create a mess" by using complex iterations. We have to admit that a "mess" might be good for hiding purposes, but only to some extent. The basic idea initiated in the paper [12,13] is of looking for potentially good Hash functions among Cayley Graphs is that girth is a relevant parameter to hashing the group $G = SL_2(F_p)$, the group of 2x2 matrices of determinant 1 over the integers modulo a prime p and $A = \begin{pmatrix} & \end{pmatrix}$ and $B = \begin{pmatrix} & \end{pmatrix}$ and theyanalysed the girth of the Cayley graph of the group.

At CRYPTO 94 [9], Tillich and Zemor, proposed a family of hash functions, based on computing a suitable matrix product in groups of the form $SL_2(F_{2^n})$.

In that paper devised the Hash Function as follows: to an arbitrary text of {0, 1}* , associate the string of {A, B} obtained by substituting 0 for A and 1 for B, then assign to A and B values of adequately chosen matrices of Heis(Z) , those could be, $\begin{pmatrix} 0 & 0 \end{pmatrix}$ and $\begin{pmatrix} 0 & 1 \end{pmatrix}$then evaluate the product associated with the string of A's and B's in the group Heis (F_p) , where F_p is the field with p elements, p being chosen large prime number [6]. The Hashed value is the computed product, that is, multiplication by 'A' or 'B' in Heis (Fp) requires essentially 9 additions, so hashing an n bit text requires 9n additions of logp bits, which is reasonably fast. In [7], using the new generators, the hash function is constructed which is also as efficient as in the provably secure hash function [6].

In this paper, we propose a new version of hash function which is a modification of a Cayley Hash Function using a Discrete Heisenberg group.

II. PRELIMINARIES

A. Definition

In [5], the vertices 'x' of the graph G and one draws a directed edge from x to y, labelled by the group element g for any x, g Є G if and only if y=xg. The group consisting of all such vertices and edges will be denoted by Cay (G, G).

B. Definition

A Hash Function h: D → R where the domain D = {0, 1}*, and the range R = $\{0, 1\}^n$ for some n≥1.

In [1,8], A One-Way Hash Function is a function h that satisfies the following conditions:

1. The input x can be of arbitrary length and the result h(x) has a fixed length of n bits.
2. Given h and an input x, the computation of h(x) must be easy.
3. The function must be one-way in the sense that given a y in the image of h, it is hard to find a message x such that h(x)= y (preimage-resistance), and given x and h(x) it is hard to find a message x' ≠ x such that h(x') = h(x) (second preimage-resistance).

[1,8] A Collision-Resistant Hash Function is a function h that satisfies the following conditions:
1. The input x can be of arbitrary length and the result h(x) has a fixed length of n bits.
2. Given h and an input x, the computation of h(x) must be easy.
3. The function must be collision-resistant: this means that it is hard to find two distinct messages that hash to the same result (i.e., find x and x' with x ≠ x' such that h(x) = h(x')).

The following definition is defined by [2,4]

C. Definition

The Heisenberg group is the group of 3×3 upper triangular matrices of the form $\begin{pmatrix}1 & a & c\\ 0 & 1 & b\\ 0 & 0 & 1\end{pmatrix}$ under the operation of matrix multiplication. Elements a, b and c can be taken from any commutative ring with identity, often taken to be the ring of real numbers (resulting in the "continuous Heisenberg group") or the ring of integers (resulting in the "discrete Heisenberg group").it is denoted by Heis group.
From this Definition, it is easily seen that the discrete Heisenberg group is generated by $\begin{pmatrix}1 & 1 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}$ and $\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 1\\ 0 & 0 & 1\end{pmatrix}$

D. Definition

In [6] Let $A=\begin{pmatrix}1 & 1 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}$ and $B=\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 1\\ 0 & 0 & 1\end{pmatrix}$ be a pair of generators of Heis(F_p) and let
$m = m_1 m_2 \dots\dots\dots\dots\dots m_n$ be abinary string. Then
$H(m) = \pi(m_1)\pi(m_2)\dots\dots\dots\dots\pi(m_n)$, where $\pi(m_i) = \begin{Bmatrix} A\ for\ i=0\\ B\ for\ i=1\end{Bmatrix}$; 0≤ i ≤ n.
This hash function is strongly related to the Cayley Graph associated with Heis (F_p) and generators A, B denoted by G̓.

E. Properties of Hash Function

Recall that the hash function construction presented above is directly associated with the Cayley graph G̓ (G, S), where G is a group generated by the elements of the set S.

1. Concatenation Property: If x and y are two texts, then their concatenation x++y is the hashed value $H(xy) = H(x)H(y)$. This clearly allows an easy parallelization of the scheme, and pre-computations when parts of the message are known in advance.
2. Parameters of the Associated Cayley Graph: We can associate to this scheme the Cayley graph (G, S): its vertex set is G and there is a directed edge from g_1 to g_2 if and only if $g_1^{-1}g_2 \in S$

III. MODIFIED VERSION OF HASH FUNCTION

A. Definition

Here we hold a different concept to form a Factorisation Problem harder. We hold an efficient way to impose limits on the type of factorisations for attacking H.
Let an integer, g>1.Let $C \in Heis(F_p) - \{I, A, B\}$ where I is the identity element of Heis (F_p). Define $\check{H}:\{0,1\}^* \to Heis(F_p)$ by $\check{H}(m) = \prod_{i=1}^{n} D_i$ where
$D_i = \begin{Bmatrix}\pi(m_i)\ if\ g \nmid i\\ \pi(m_i)C\ if\ g \mid i\end{Bmatrix}$ and $m = m_1 m_2 \dots\dots\dots\dots\dots m_n$.

B. Security Properties: First, we say that the $\check{H}$ is as secure as H

1. Proposition: If we know a message b such that H(b) = C, then we can efficiently find a message x', for every message x such that $\check{H}(x) = H(x')$.

Proof:
Let b be a bit strings such that H(b)=C.
Let $x = x_1 x_2 \dots\dots\dots x_n$ and assume $x' = x_1 x_2 \dots\dots\dots x_g b x_{g+1} \dots\dots\dots x_{2g} b x_{2g+1} \dots\dots\dots x_n$
Then we can simply say that $\check{H}(x) = H(x')$.

2. Theorem: Breaking the preimage and collision resistance of $\check{H}$ respectively leads to breaking the preimage and collision resistance of H.

Proof:
Let $C \in Heis(F_p) - \{I, A, B\}$ then $\check{H}(x) = C$.
By above proposition, there exist ḿsuch that $H(x') = \check{H}(x) = C$.

Hence it says that Preimage Resistant of $\check{H}$ gives the Preimage Resistant of H.
Thus, the first part of the theorem holds.
Second Part of the theorem proves that, let $x \neq x'$ with $\check{H}(x) = \check{H}(x')$
By above proposition, there exist ḿ and ḿ' such that $\check{H}(x) = H(\acute{m})$ and $\check{H}(x') = H(\acute{m}')$.
So, $H(\acute{m}) = \check{H}(x) = \check{H}(x') = H(\acute{m}')$
Hence Proved that the collision for $\check{H}$ is also a collision for H

IV. EFFICIENCY

Suppose we take an input binary strings of length n with a prime number p (is of order 2^256) and fix g (for example g = $\lfloor \log_2 (p) \rfloor$-1) and we use $\mathbf{C} \in \mathbf{Heis}(\mathbf{F_p}) - \{\mathbf{I}, \mathbf{A}, \mathbf{B}\}$ as fixed matrix (or we can assume C as one of the smallest hashed value of binary strings of length l<n).

Using these above parameters, we can compute the hashed value of Ȟof length nwith$\lfloor n/g \rfloor$ matrix multiplications which is less than H, since it has 3n matrix multiplication That is, the performance of the defining Hash function is not increase too much as the Cayley hash function defined in [6].

V. CONCLUSION

Cayley Hash function constructed from Cayley graphs is very effective. We described a new Cayley Hash function where '0' and '1' bits are hashed by Discrete Heisenberg group over F_p. We proposed the pair of generators

$$\boldsymbol{A} = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix} \text{ and } \boldsymbol{B} = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}.$$

Our proposal is very efficient and it is possible to compute the hash function of Ȟ of length n with $\lfloor n/g \rfloor$ matrix multiplications which is less than to H, since it has 3n matrix multiplication in [6]. Hence performing $\lfloor n/g \rfloor$ matrix multiplication in F_p, where p is a large prime to hash a bit string of length n. In addition, the calculation of any Cayley Hash function can be parallelized easily, which can improve our proposed hash function efficiency. Hence we conclude that our new defined hash function is efficient and secure as the hash function which is defined [6]. 410

REFERENCES

[1] Bart Van Rompay, "Analysis and Design *of* Cryptographic hash Functions, MAC algorithms and Block Ciphers*", Doctoral Dissertation, KU Leuven, D/2004/7515, June 2004*.

[2] Capogna Luca, Donatella Danielli, Scott D. Pauls, and Jeremy Tyson, An Introduction to the Heisenberg Group and the Sub-Riemannian Isoperimetric, *Springer Science & Business Media, Progress in Mathematics*, Aug-2007.

[3] R. Daugles Stinson, "Cryptography theory and practice, Second Edition", *Chapman & Hall/CRC Press, Febuary-2002*.

[4] C. Hall Brian. "Lie Groups, Lie Algebras, and Representations: An Elementary Introduction" *Second Edition, Berlin: Springer International, 2004.*

[5] A. Joseph Gallian, "Contemporary Abstract Algebra, 8thEdition", *University of Minnesota, Duluth*, ISBN-10:1133599702, ISBN-13:9781133599708, 2012

[6] V. Kochamani Vibitha, P.L Lilly, and K.T Joju, "Hashing with Discrete Heisenberg Group and Graph with large girth", *In Journal of Theoretical Physics and Cryptography*, Vol. 11, pp 1-4, May 2016.

[7] P. L. Lilly and V. Vibitha Kochamani, "Hashing with Discrete Heisenberg Group using New generators", *In Journal of Theoretical and Computational Mathematics*, Vol. 2, No. 2, pp 14-18, Nov 2016.

[8] B. Praneel: "Analysis and Design of Cryptographic Hash Functions", *Doctoral Dissertation in K. U.Leuven* Jan. 1993.

[9] V. Shpilrain, "Hashing with polynomials", *Lecture Notes in Computer Science. Springer*, No. 4296, pp.22–28, 2006.

[10] Tillich Jean-Pierre and Gilles Zémor, "Group theoretic hash functions", *Proceedings of the First French- Israeli Workshop on Algebraic Coding (London, UK), Springer-Verlag*, pp. 90–110, 1993.

[11] J. P. Tillich and G. Zemor, "Hashing with SL2", *Advances in Cryptology Lecture Notes in Computer Science, Springer-Verlag*, Vol .839, pp. 40-49, 1994.

[12] ZémorGilles, "Hash functions and Cayley Graph", *In Designs, Codes and Cryptography, Springer*, Vol. 4, No. 3, pp. 381-394, July 1994.

[13] ZémorGilles, "Hash functions and graphs with large girths", *In Eurocrypt, Lecture Notes in Computer Science, Springer*, Vol. 547, pp. 508-511, 1991.

[14] Xwang, Y.L. Yin and H.Yu, "Finding collisions in the full SHA-1" *In CRYPTO, .Lecture Notes in Computer science, Springer*, Vol. 3621, pp.17-36, 2005.